Linux:

1964—-------bell labs(New jersey)
1968-69 —------
1970 —------dennis rechie and ken thomson ee project continue
UNICS— uniplexed information and computing services
1975 —----- unix v6

Ibm —aix
Sunsystem- solaris
Apple- macos
HP- ux

Unix —---linux

Unix is nothing but linux / linux is a clone of unix

In 1995 —-linus torvald
Andrew tannenbaum is a professor —------minix OS

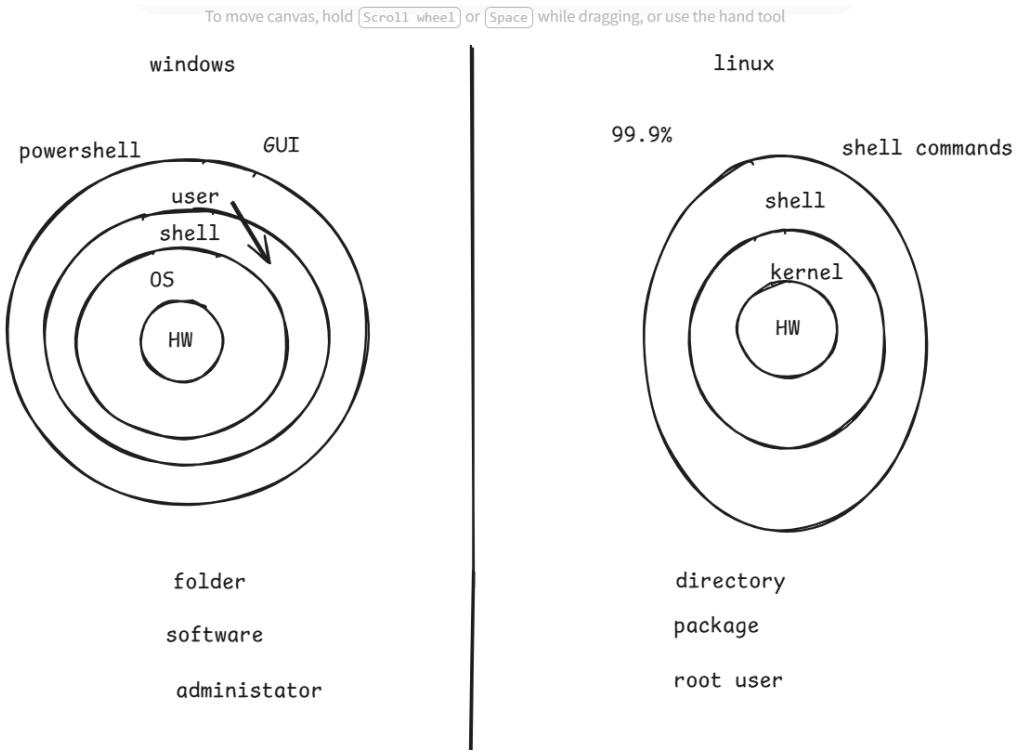
MINIX and UNIX OS refer—------linus torvald —----------linux develop

Linux —-------REDHAT, centos, debian, Ubuntu, amazon linux

OS = CLI and GUI
Cli- command line interface ex: LINUX
GUI- graphical user interface ex: windows

Linux features:

Opensource
User friendly
secure

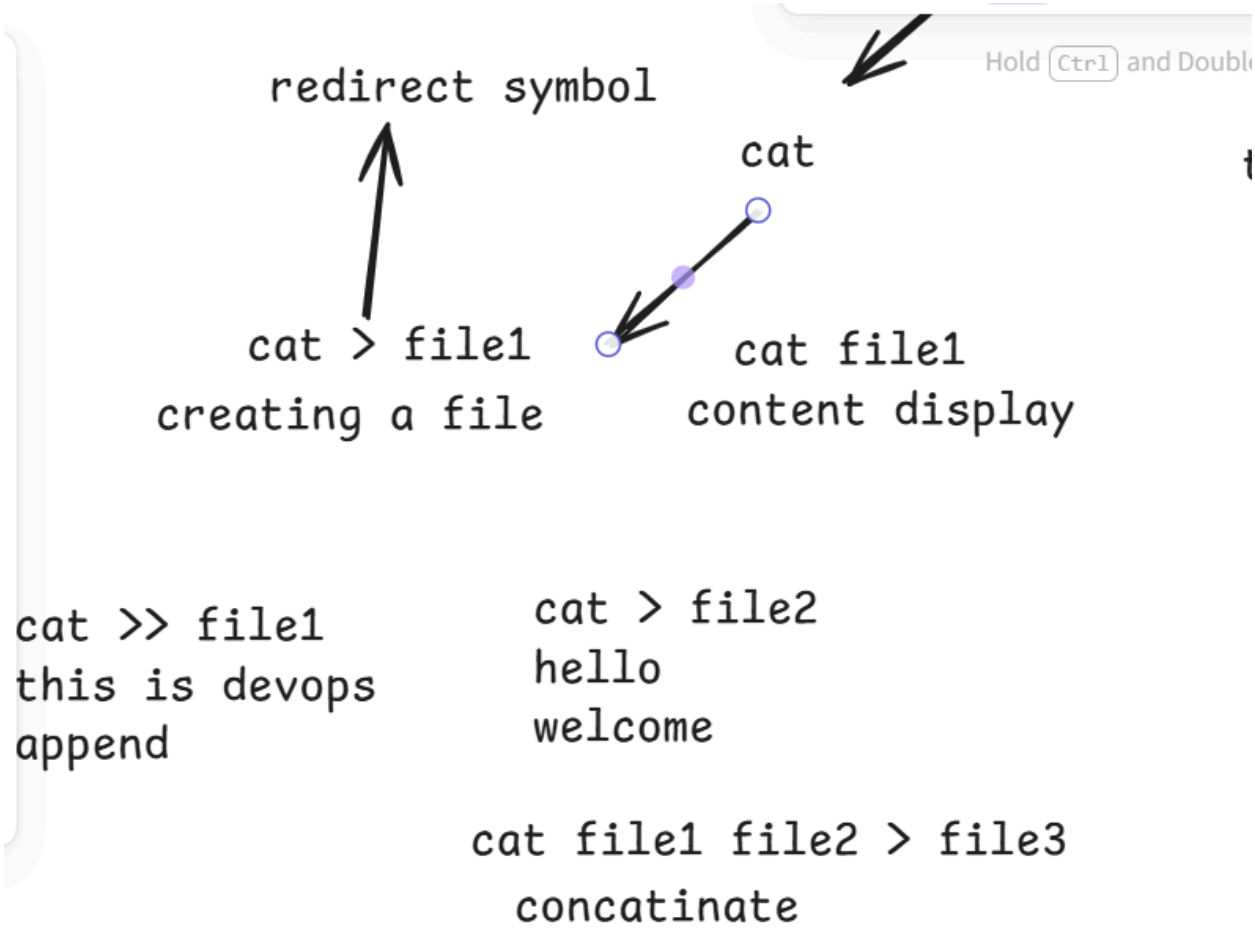


File system Hierarchy:

/root - Administrator . It is a root directory for root user
/home - home directory user created by root user
/USR - default application or software install
/Boot- bootable files will present
/etc - configuration files
/bin: history of the root user/ commands used by root user or other user
/sbin: commands used by only root user

Basic commands:

Root : sudo su - (super user do switch user { - login })
Ls: list all the files and directories
PWD: it will show our present working directory
CAT:



Touch: create a empty file
Create a multiple files
While creating it will create time stamp also

Nano and vi editor: create a file and edit the content

Echo : create a file
Print the content
Print the multiple content

Mkdir: create a directory
Create a multiple directories at a time

```
1  ls
   2  pwd
   3  clear
   4  cat > file1
   5  touch file2
   6  ls
   7  cat file1
   8  cat file2
   9  touch f1 f2 f3
  10  ls
  11  stat f1
  12  ls
  13  vi f1
  14  cat f1
  15  nano f2
  16  cat f2
  17  vi f4
  18  ls
  19  cat f4
  20  nano f5
  21  ls
  22  cat f5
  23  vi f6
  24  ls
  25  echo "hello" > f6
```

```
26  ls
27  cat f6
28  echo "swathi"
29  echo "hello all"
30  echo -e "good morning \t welcome"
31  echo -e "good morning \n welcome \t to devops"
32  ls
33  mkdir d1
34  ls
35  cd d1
36  ls
37  cd ..
38  mkdir d2
39  cd d2
40  cd
41  ls
42  mkdir -p dir1/dir2/dir3
43  ls
44  cd dir1
45  ls
46  cd dir2
47  ls
48  cd dir3
49  ls
50  pwd
51  cd ..
52  cd dir1/dir2/dir3
53  pwd
54  cd
55  cd dir1/dir2/dir3
56  cd ../../..
57  cd dir1/dir2/dir3
```

```
 58  cd ../..
 59  cd
 60  mkdir -p m1/m2/m3/m4/m5
 61  ls
 62  cd m1/m2/m3/m4/m5
 63  pwd
 64  cd ../../../..
 65  cd
 66  history
```

Copy: copy the file content to another file
Copy the file from one directory to another directory

Move: mv the file/directory from source to destination

Rmdir: removes the empty directory
Rm -rf(recursive force) : remove the file/directory

Note: don't use (rm -rf *) to delete. Cant be recoverable again.

Permissions:

-rw- r-- r--

r ----read = 4
w-----write = 2
x-----execute = 1

rwx rwx rwx

user group others

-rw- r-- r--

rw- = 4+2 = 6
r-- = 4 = 4
r-- = 4 = 4

644

rwx = 4+2+1 = 7
rwx = 4+2+1 = 7
rwx = 4+2+1 = 7

777

rwxr--r-- = 744

r-x-wxr-x = 535

--xr---w- = 142

2nd method of giving permissions

ex: rw-rw-r--

chmod u+x,g+x,o+w+x filename

r----x-w-

chmod u+w+x,g+r+w,o+r+x filename

rwxrwxrwx

user= read
group= write
others= execute

chmod -R u-w-x,g-r-x,o-r-w filename

r--r-xrw-

chmod -R u+w+x,g+w,o+x filename

Link:

2 types:

Soft link: ln -s filename linkname
It is a shortcut to the file. If we make changes to the original file it will reflect in link file. If the original file gets deleted then link file doesn't work.

Hard link: ln filename linkname.
It is like normal file. It is a shortcut to the file. If we make changes to the original file it will reflect in link file. If the original file gets deleted then our link file still work.

Port numbers:

65535 ports

Tar:
tar -xzvf

-z = decompress .gz files
-x = extract files from archive
-v = verbose mode(show the files being extracted)
-f = specifies the file name

GIT Tool:

Git is a version control tool. Git is used to keep the track of the version of files and directories.

Git is open source tool.

SCM- source code management or software configuration management

1. CVCS: centralized version control system

2. DVCS: Distributed version control system

Working directory/working copy/workspace:

It is place where developers write the code

Repository:

It’s a kind of folder where codes are stored.

steps:

1. launch ec2-instance & connect
2. Install git
3. create direct and switch to direct
4. user configuration
5. git init ------initialize the git and create the 3 logical separation
6. create 1 file and add some content
7. git status ----------command will help to check our file are in which stage
8. git add . ----------it will add the files from workspace to staging area
9. git status
10. git commit -m "message"---------commit the file from staging area to local repo
11. git status
12. git log----------will get to know/list the commit ID
13. git log --oneline----------list the commit ID in a single line
14. git show commitID----------it will display the commit ID
15. git remote add origin githubrepoURL----------establish a connection b/w local repo and github repo
16. git push origin master/main
    username: XXXXX
    password: github token

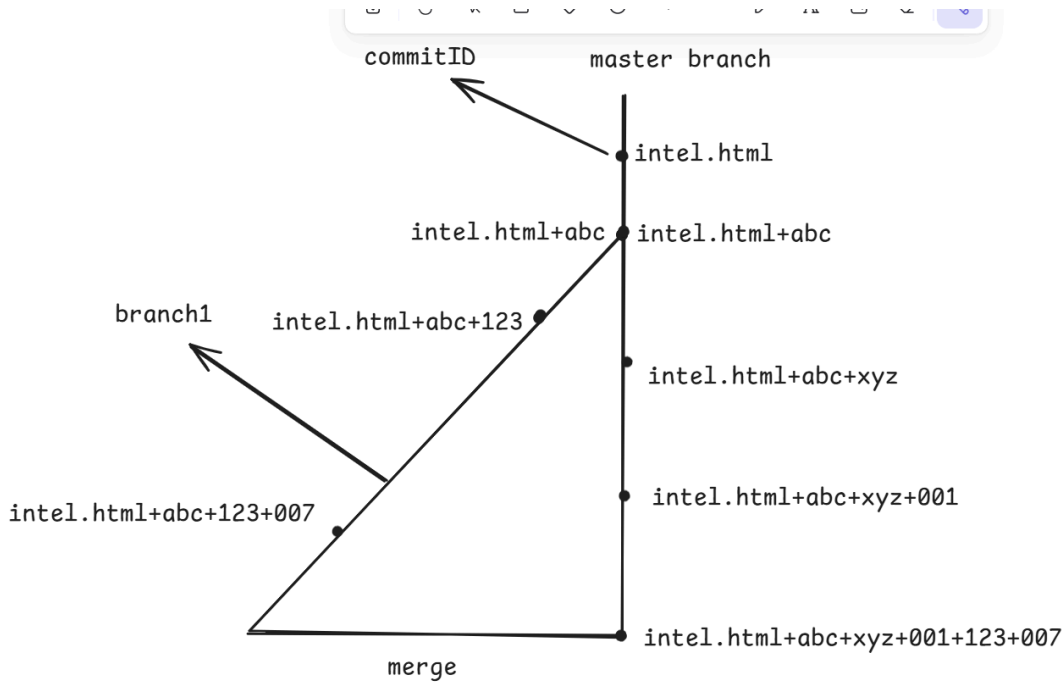
How to generate token:

github----github settings-----developer settings on left side-------personal access token----token classic---
generate token(classic)------authentication of the user------token name---------select repo-----click on

generate token

Branch:

Branch is for parallel development. If two team or people will work on same piece of code by creating 2 different branch and integrating by merging it.



Git conflict / merge conflict:
When same name file having different content in different branmches, if you do merge, merge conflict will occur

Git stash: it is a temporary storage

Git reset
Git revert

Tag
Git clone
Git fork
Git fetch
Git rebase

Jenkins

Jenkins is an open source CICD tool. CI- continuous Integration
CD- continuous deployment / continuous delivery

Plugins: logical plugin which we configure in jenkins

Pipeline: It is a job

Job: Jobs are nothing but specific way to do a task

In jenkins we have different jobs:

Freestyle
Pipeline

Build triggers:

1. Manual : Build now
2. Build periodically : It will trigger the job based on time. Whether developer push the code or not
3. Poll SCM: It will trigger the job based on commit. If there are any commit within that specific time it will trigger the job
4. Webhook : github and jenkins connection establish

5. Build after other projects are built :
6. Build remotely : token generate

crontab: Nothing but scheduler

* * * * *

The cron expression is made of five fields. Each field can have the following values.

*	*	*	*	*
minute (0-59)	hour (0 - 23)	day of the month (1 - 31)	month (1 - 12)	day of the week (0 - 6)

Pipeline:

Pipeline is where we put all our job configuration to achieve multiple instance execution in a single pipeline

Groovy: Groovy is a term used to define a new pipeline as a script

2 types:

Declarative pipeline
Scripted pipeline

Agent: It is a block to define which job to run on which machine

Master - slave arachitechture

Agent any: It is used to define at a pipeline level. It can be run on any slave based on load

Docker:

Docker is a containerization tool.

container is a virtual machine

Microservices architechture

Hypervisor: It is a tool which converts physical resource into virtual resource

Docker is an open source centralized platform designed to create, deploy and run the application

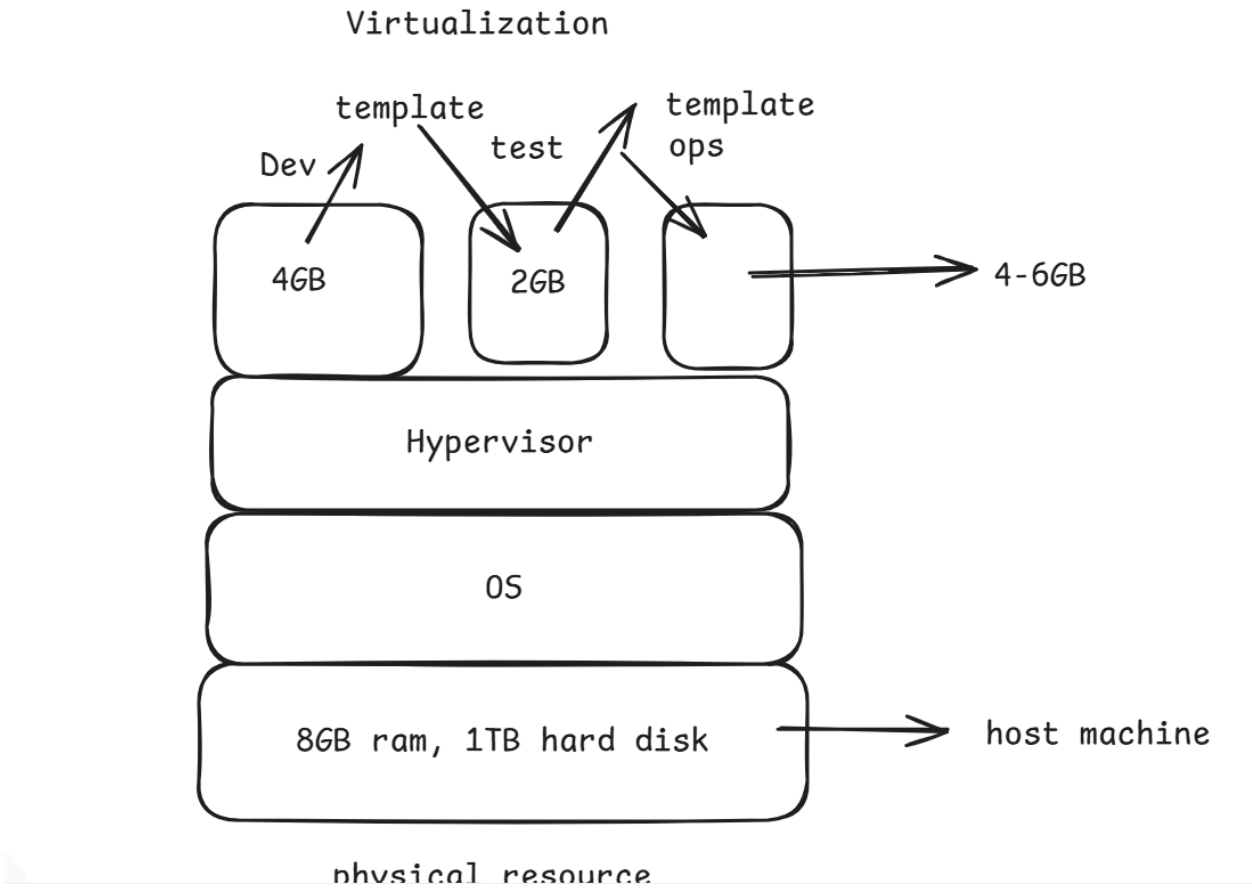
Docker image Images are read binary templates used to create docker containers

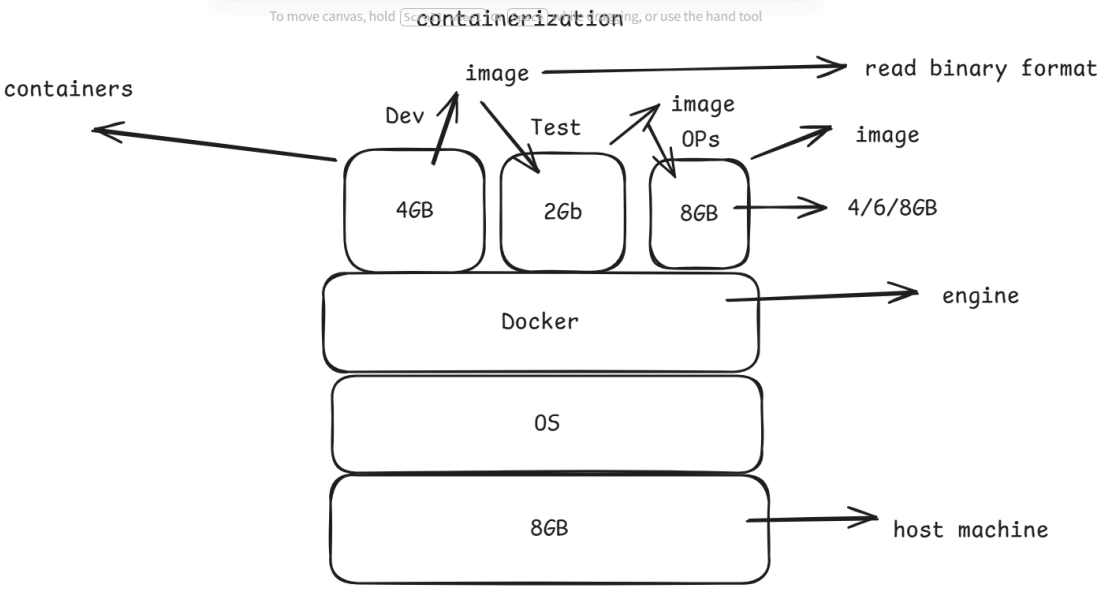
Or

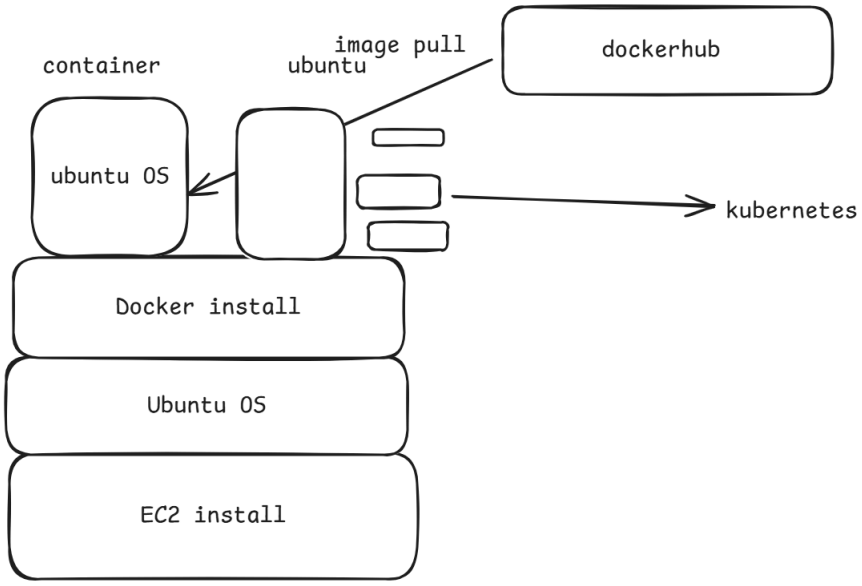
Single file with all dependencies and configuration required to run the program

Docker Containers container hold the entire packages that is needed to run the application

Dockerhub is a registry. It is a place where all images are stored


Virtualization
template
Dev
test
template
ops
4GB
2GB
4-6GB
Hypervisor
OS
8GB ram, 1TB hard disk
host machine
physical resource


containerization
containers
image
read binary format
Dev
Test
image
OPs
image
4GB
2Gb
8GB
4/6/8GB
Docker
engine
OS
8GB
host machine



docker images
docker pull imagename
docker run -it(interactive terminal mode) --name containername ubuntu(image) /bin/bash
cat /etc/os-release
docker ps(process status) -a(all) - all containers states
exit - come out of container. stop state
docker ps - container which are running
docker start containername
docker attach containername
docker rm -f container-name/cont-ID
docker rmi -f image-name/image-ID

What are the different ways to create image?

First method to take the image from dockerhub.
Second method create a image from existing container
Third method by using a dockerfile

Dockerfile: It is basically a text file. It contains some instructions. Using these instructions will create images.

Steps:

1. Create a file name Called Dockerfile(where D is capitol)
2. Add instructions in Dockerfile(where instructions should be in capitol)
3. Build Dockerfile to create the images
4. Run the docker image to create a Docker container.

Instructions:

FROM: to pull the base image or Container OS
RUN: it is used to execute the command
WORKDIR: It is used to set the working directory for a container
COPY: it is used to copy the normal file from local ,machine to container machine
ADD: it is used to copy the normal file/zip file/tar file/compressed file from local ,machine to container machine
CMD and ENTRYPOINT: it is used to execute the command but during the container creation
EXPOSE: to specify the port details of the package

Docker storage

3 types:

Docker volume : It is basically a persistent storage. It stores the data in the docker own place. Even if the container gets deleted our data will be safe.
This is managed by the docker completely.

docker bindmount: It is not managed by docker. It is like sharing the folder from our laptop/host machine to the container machine.

docker tmpfs: It is a temporary storage. Once the container will go the stop/exited state, our data will be destroyed.

Port mapping:

It tells docker daemon to associate port of the container to the port of the host machine

Docker compose:

It is a tool in docker for defining and running multiple containers docker application. We compose yaml file to configure our application services. Then with single command we create and start all services from our configuration

Kubernetes

Kubernetes is a orchestration tool.

Orchestration is nothing but management. To manage multiple containers deployed on different host machine.

K8s is an open source container orchestration tool. Means in order to maintain multiple containers we use kubernetes

Cluster: it is a group of nodes.(master as well as slave)

K8 is master-slave architecture.

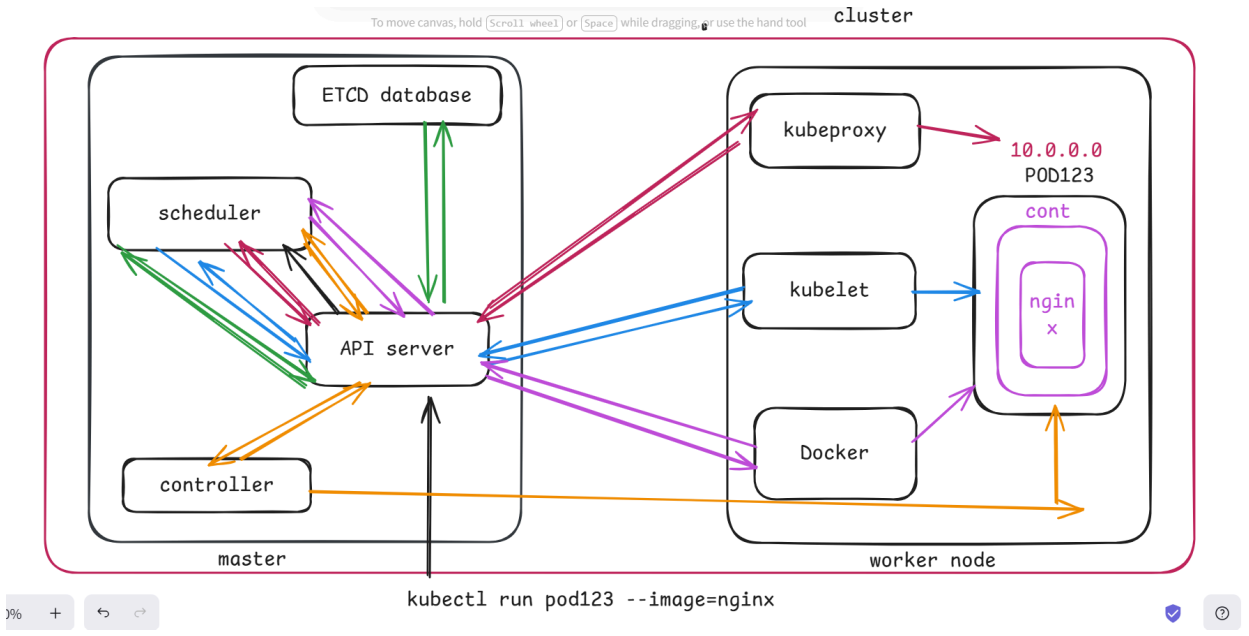
Kubernetes components:

1. API server
2. scheduler
3. controller manager
4. etcd database
5. kubeproxy
6. kubelet
7. container ---docker
8. POD

IMP note: In k8s containers always created under a logical boundary. That logical boundary is called POD

IMP note: every components will communicate via API server.

Kubernetes Architechture:



Ansible

It is a open source configuration management tool.
Ansible is a master-slave architechture
2012-----michael Dihaan. Red hat acquire.
ansible is a agentless server

Inventory:

It is a file where we store all the information of other system.

Ansible Playbook:

It is a file where we write our logic to install any packages.
playbook is nothing but yaml file.

playbook terms:

1. Hosts: Target machine
2. Tasks: steps to install the package
3. Modules: built in actions. apt, service, copy
4. Become: sudo privilages

Terraform:

Terraform is a tool which is used to automate the infrastructure.
with the help of terraform we can create and destroy the infra in a very short period of time

IAC tool(Infrastructure as a code)

HCL- Hashicorp configuration language

Provider: Providers are required for your configuration files to work.

Terraform init : Initialize the terraform
Terraform plan: plan generate
terraform apply: it will create a resource
terraform destroy: resource will be deleted

ec2-----aws_instance
vpc-----aws_vpc
subnet---aws_subnet
S3---aws_s3

Maven:

Maven is a build tool to run the java projects
Maven go life cycles

mvn validate----validate the source code
mvn compile----source code compile
mvn test-----test the compiled source code
mvn package----pack the compiled source code in distributed format to create the artifact
mvn integration-test----- jar/war integration test
mvn verify-----verify the quality of the jar/war file

Prometheus and grafana

It is a open source tool. it is used for monitoring the metrics.

Defi: Prometheus is one of the linux server monitoring tool. It is a open source tool. It is used to monitor the metrics

Defi: Grafana is a free and open source visualization tool. Mostly used with prometheus for monitoring the metrics.